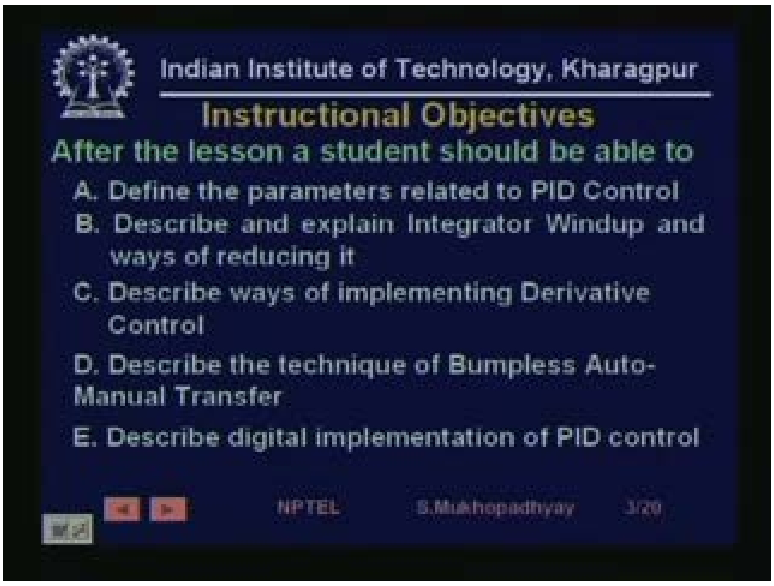
Industrial Automation & Control
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 12
P-I-D Control

Good morning and welcome to lesson 12 of this course on PID controls. So, as usual before starting the course we will review the instructional objectives.

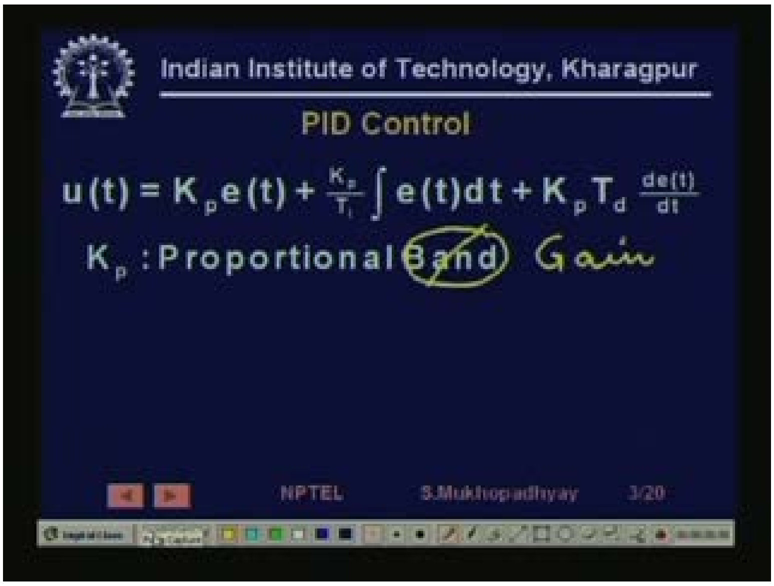
(Refer Slide Time: 01:02)



And these are. Firstly, that we will be related we will be, we will learn how to define the related parameters of PID control in an industrial context. Secondly, we will describe and explain in detail about a phenomenon, which may times occurred with PID control known as integrator windup, and the ways of reducing that we will describe various ways of implementing. The derivative control part we will also describe the one technique of you know bump less auto manual transfer that is when the control is transferred from auto to manual or manual to auto, how?

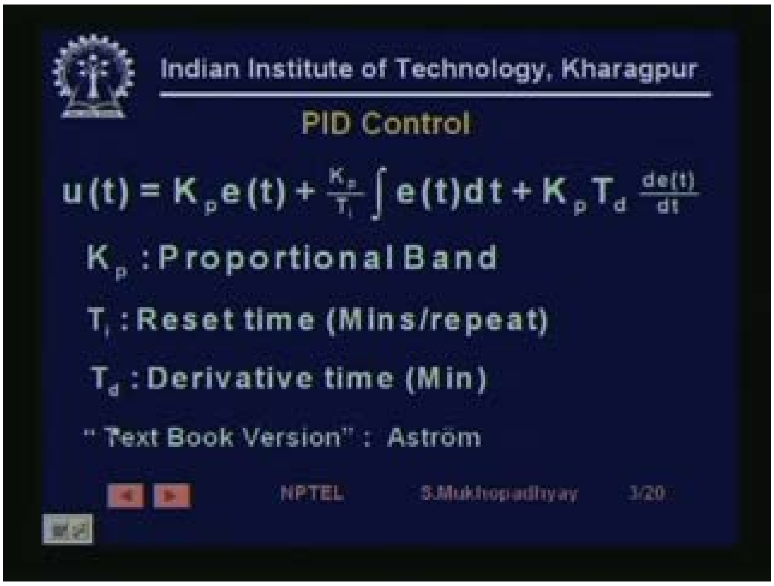
So, that it can happen without any short to the process, and finally, we will describe digital implementations of PID control. So, in other words we are going to look at various practical aspects of PID control today. So, let us begin with the PID equation.

(Refer Slide Time: 02:12)



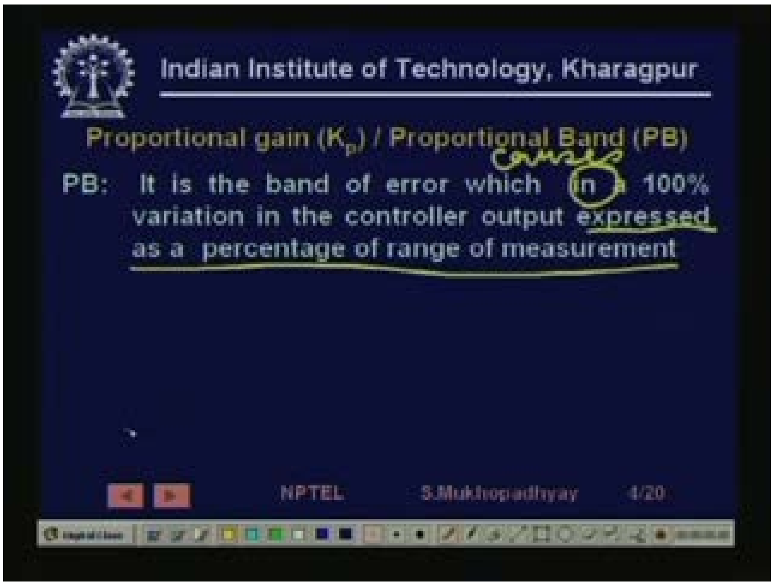
This is the PID equation, which we have seen in the last lesson also, where K p is the proportional gain or sometimes we this is not proportional band as written, but it is, it is proportional gain. But we will, but a very similar parameter called proportional band is also used in the context of PID controllers, we will see soon how it is related to the proportional gain.

(Refer Slide Time: 02:53)



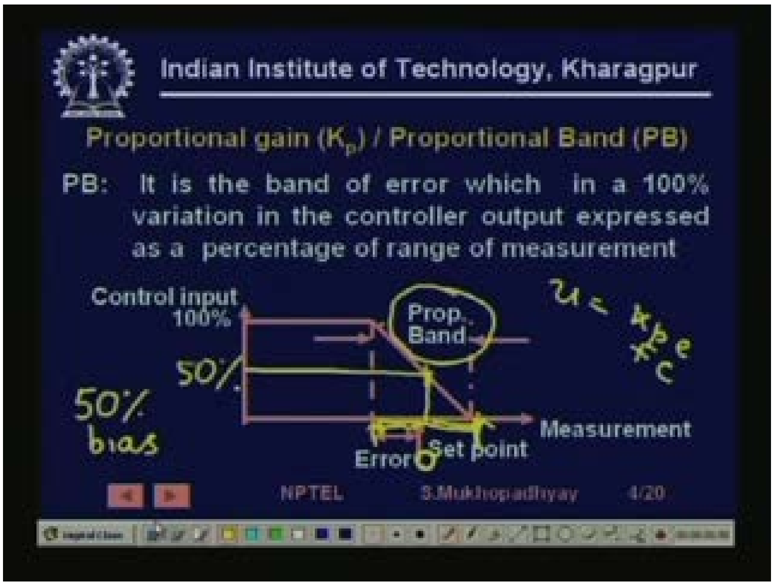
Next, is the parameter T i the parameter, T i here, which is called the reset time and expressed in a peculiar sounding unit called minutes per repeat. Next is the derivative time. So, derivative time here note the, note the time units of minutes. These are rather unusual it may seem rather unusual, but remember that typical chemical processes have time constant of the order of the minutes. So, these times are often expressed in minutes. This is as very well known control scientist Karl Johan Astrom says also called text book version of the PID control equation. As we will see in this, in this lesson that there are various modifications that you have to do to this equation before it can be implemented. So, let us first go about defining the various terms. So, we first define proportional gain or proportional band.

(Refer Slide Time: 04:02)



Proportional gain is well known it is K p, which is delta u by delta e or u by e, while proportional band this term is new, and it is defined in just inverse way. So, proportional band is defined as it is the band of error, which in a 100, which causes, which causes a 100 percent variation in the controller output and generally expressed as a percentage of the range of the measurement. So, that is the definition. So, it is in an inverse way where gain is u by e, here we are defining P B as the band of error, which causes a 100 percent variation in the controller output or the manipulated input to the plant right. So, in that sense it is a, it is a inverse of K p.

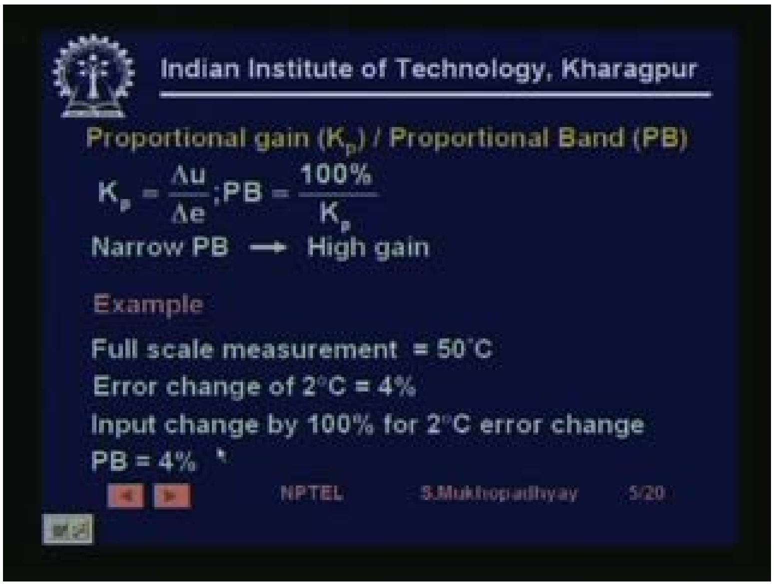
(Refer Slide Time: 05:12)



So, look at this diagram will clarify matters further. So, here look at the controller input and suppose this is the set point currently the set point is set here. So, if the measurement or the output is the measurement or the output could be anywhere in this zone. So, it will cause various kinds of error, and if we use a proportional band. Then as the error will increase the output will increase in this case the proportional controller actually has a 50 percent bias, which means that when the error is 0 there is still a 50 percent output of the controller, sorry this line is getting.

So, this is typically set at 50 percent. So, there is a. So, the controller output equation is actually given as U is equal to K p into e plus a constant term. So, plus a constant term c and this constant terms is actually 50 percent. So, when the e is 0 still you get 50 percent output will otherwise there will always be a steady state error as we have seen in the last lesson. So, what happens is that as the error changes to this side or to this side the output decreases or increases, and if the error changes from here, from here to here the input to the plan increases from 0 percent to 100 percent. So, this is the band of error, which causes the, causes an output a variation in the controller output from 0 percent to 100 percent, and this is the proportional band right. So, look at let us look at an example.

(Refer Slide Time: 07:37)

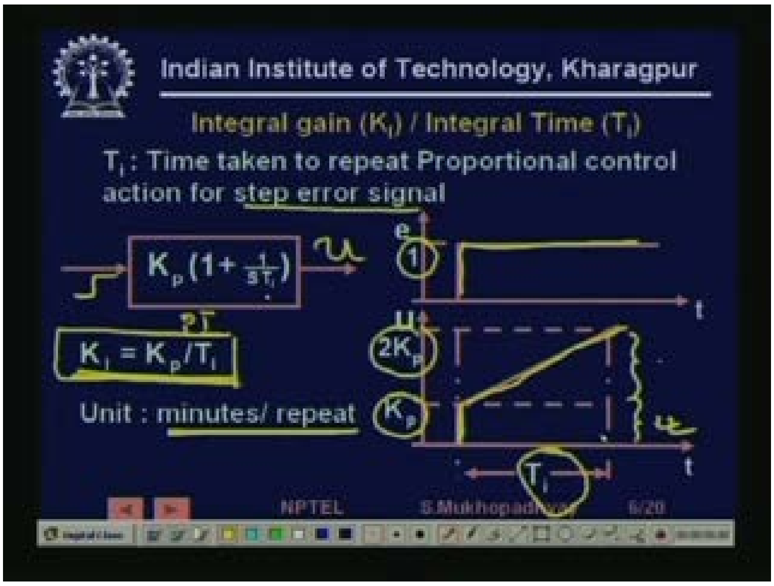


So, while K p is delta u by delta e proportional band is defined as 100 percent by K. So, you can easily find out that this gives the error in percentage, which will cause a 100 percent input change. Obviously, a narrow K p means low value of K p implies a high value of K p. A narrow P B or a low value of the proportional band implies a high proportional gain right.

So, let us look an example suppose, the we are, we are, we are talking about a temperature control loop, where the full scale measurement is 50 degree centigrade right. Suppose, an error of 2 degree centigrade, which is 4 percent of 50 degree centigrade causes an input change by 100 percent. So, maybe there is a heater, who whose output will change from 0 watt to 5000 watt or 1000 watt or whatever.

So, if the error changes by 2 degree centigrade, then the heater output will change from 0 percent to 100 percent. So, in such a case 4 percent change in error causes a 100 percent change in input. So, the proportional band in this case is 4 percent. So, this is the meaning of the proportional band.

(Refer Slide Time: 09:12)



Now, let us look at the integral gain, which is again expressed in terms of the integral time and the proportional band. Now, here I would you might think that why is it, why is it that rather than expressing the integral gain rather than expressing the integral gain as K I, why I am expressing it as K p into T I, why the derivative gain, which I could call K b. I am, I am expressing as K p into T d, what is the, what is the reason?

The reason is that the reason is actually embedded in history turns out that in the older. You know hydraulic and pneumatic PID controllers, the construction of the controller was such that one part of the device used to control K p, another part of the device used to control T i another part of the part of the device is used to control T d. So, there are certain distinct parts of the controller, which used to realize these terms K p T i and Td.

So, that an average. So, that an overall integral gain of K p by T i and an overall derivative gain of K p into Td is realized. So, it is for from that principle that the integral time and the derivative time terms are continuing, but if you have a if you, if you have a microprocessor based controller then all these terms need not be considered. Then, you could equivalently work with you know K i and K d, but still let us since I mean, since the terminology continues. So, let us see the meaning, because K i and K d we understand very well they are just simply the gain terms.

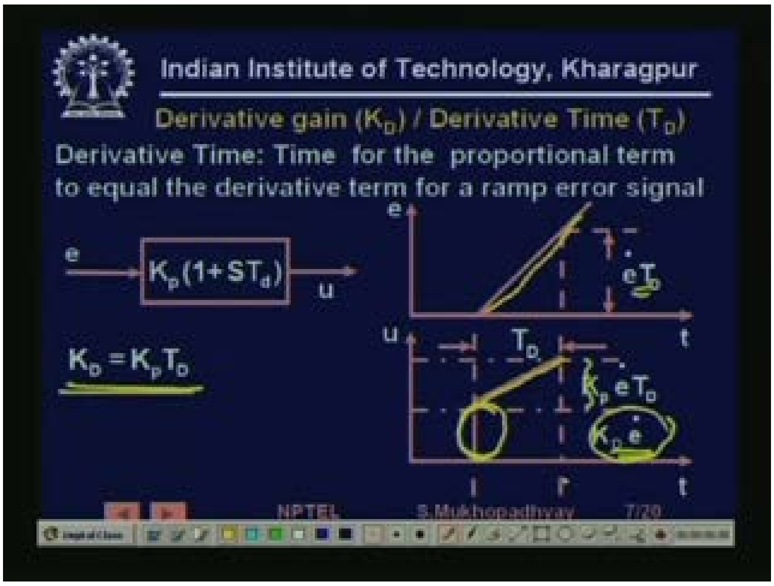
So, let us see, what is the integral time? So, integral time is the time taken to repeat the proportional control effort or action for a step error signal. So, what happens is that let us this probably not. So, clear. So, let us look at the, let us look at the scenario. So, now suppose take this as a, this is a P I controller right. This is the P I control controller. Suppose, we are giving a step error signal to each.

So, here we have a I am sorry. So, here we have a step input, if you give a step input like this, like this to the controller then how will the u output vary. The u output will vary like this. So, immediately the K p part will rise. So, this will be K p into e, and then the integral term will start integrating the error. So, it will go up right and after some time after sometime this integral part of the input will equal the proportional part of the input

So, the, so it turns out then after exactly after T i amount of time the input will become, if the proportional control input is K p because I have taken the error as unity. Then, after T i amount of time the total input will become 2 K p or the integral part will repeat the proportional part. So, in that sense this definition is now explained that is time take this time taken to repeat the proportional control effort action for a step error signal.

It is given as we all know it is given as K p by T i the proportional, the integral gain is expressed as K p by T i. Now, so from this definition perhaps it is now clear why it is expressed as minutes per repeat. So, for if the, if this continues then every T i minutes the integral terms will produce another K p times input right. So, the proportional control it will continuously repeat every T i minutes in that sense the unit is minutes per beat. So, that is the that explains the integral term. Now, we go the derivative term again.

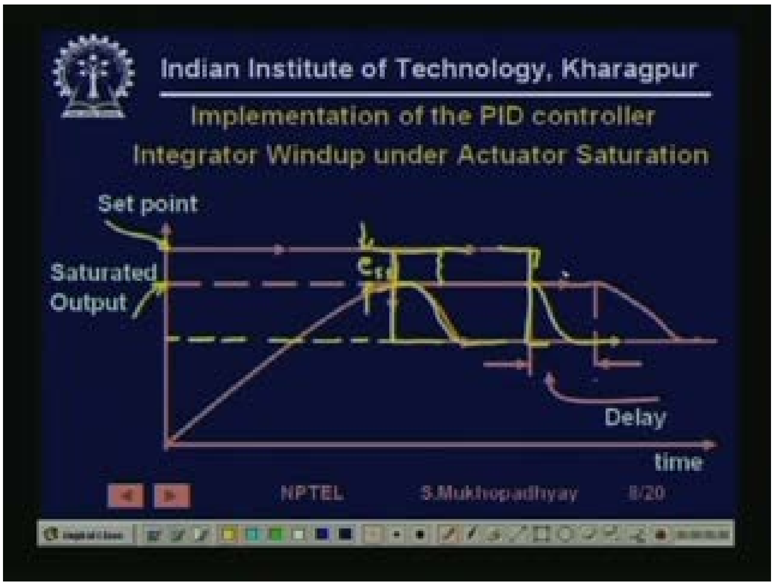
(Refer Slide Time: 13:38)



We have a derivative gain and we have a derivative time. So, again now the derivative time is the time taken for the proportional term to equal the derivative term for a ramp error signal. So, again a similar thing. So, let us look at the diagram here. So, here now we have a P D controller right. So, in the P D controller, if you fit with a ramp signal, now let us say a ramp signal of some of some slope e dot some slope e dot then immediately what will be the, what will be the. Now, the now, the now the derivative term will jump because there is a constant e dot. So, there will be immediately a K D into e dot term, and the K p term will now start going up because e is going up.

So, after T D time this is going to be K p e dot into T D. So, if K p e dot into T D has to equal K D into e dot, which is the derivative term output then K D equal to K p into T D. So, this is the time, this is the time or this is the derivative time after which the proportional action will repeat the derivative action. So, that explains, what is the meaning of the term derivative time.

(Refer Slide Time: 15:06)



Now, we come to the implementation of the PID controller, and we basically we are trying to see that what are the problems that might occur if you just simply implement the term as a proportional as an plus integral plus derivative. So, first we look at the integral term in detail, and see what happens when there is actuator saturation. Now, you see actuator saturation is actually very common in the sense that only in certain cases see the set point keeps on varying.

So, suppose the, suppose the set point stays suppose the set point stays 80 percent of the time. It stays in about 60 percent of its maximum value and probably a 5 percent of the time it reaches something like it reaches 100 percent. Now, if you have to make an actuator, which can really deliver full output even for a 100 percent control input completely proportional. Then the, then the actuator has to be very large and the, and the actuator setting has to be I mean the actuator power rating has to be very large.

So, often it is it is very common that we will we will chose an actuator which can deliver input proportional to the control input for about 75 percent then it will saturate. So, the case is where you will, you will get you are going to give very rare cases sometimes very exceptional cases may be you will give more than 75 or 80 percent and that time there will be some error, and you are willing to tolerate.

So, this happens in many cases now. So, we want to see what happens to the PID controller in such cases of actuator saturation. So, let us look at this case very carefully. So, you see that suppose the maximum possible output that the actuator can produce is this. So, here it saturates it cannot produce any further output, but a set point is given, which is higher than that. So, the actuator naturally cannot give enough input corresponding to this set point.

So, what will happen is that the output will rise and then here it will saturate, it cannot. So, the output cannot increase beyond this point and this amount of error this amount of error will in, will exist this is the steady state error, which will exist. One cannot do anything about it simply because whatever control you apply the actuator will not be able to give input. So, the plan to input will not increase beyond this that is fine.

Now, suppose the set point is reduced here, it is you have realized that it cannot reach that set point. So, it is reduced. So, immediately now this output level, this output level is very much reachable by the actuator. So, what is desirable is that the actuator will immediately because by control action the output will come down will reach at 0 steady state error point as is common under integral control, but exactly that does not happen, why it does not happen?

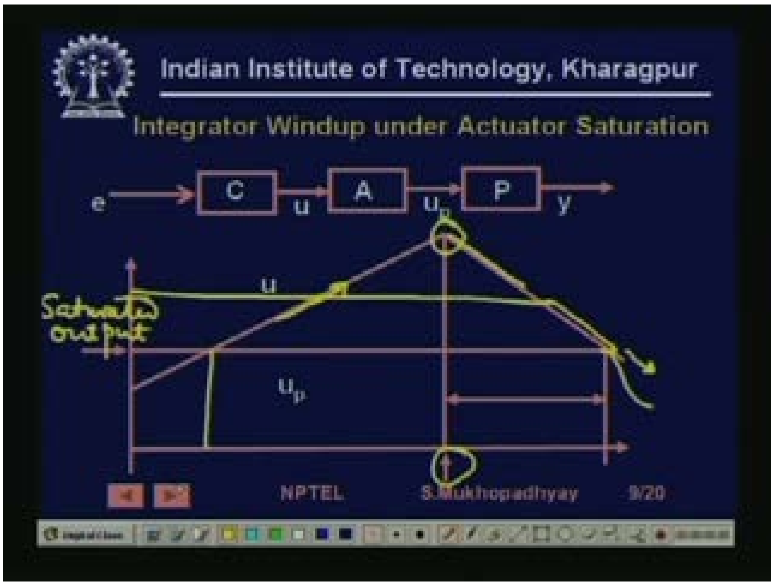
Now, suppose that you have held this error you have not immediately reduced it, but you have continued with it for some time. So, now what is happening here during this time the error is constant, the error is constant. So, the proportional term. So, the proportional part of the control input remains constant, but the integral term of the control input goes on increasing. So, the integral in the PID controller goes on integrating the error.

However, it cannot produce a control input because the that input is given. So, the as if the PID controller, output is the controller, output is continuously increasing, it is also coming the actuator input, but the actuator is not able to give that output because it is already saturated.

Now, suppose that after some time the control input is now reduced. Now, what is going to be happen, what will be observed is that the while, the while it would have been desirable. That the control input immediately falls down and reaches the desire steady state point. It does not do that rather it continues at the same level ignoring that the, that the, that the set point has now been reduced, and after some time, only after some time does the actuator, does the control start respond to the set, to the set point, and this phenomenon is called integrator windup.

Basically what has happened is that the integrator has become bloated floated. So, it has not realized that the plant cannot reach this output. So, the error is will persist. So, it is unnecessarily trying to give more and more control input and getting blown up right. So, that is integrator windup.

(Refer Slide Time: 20:35)

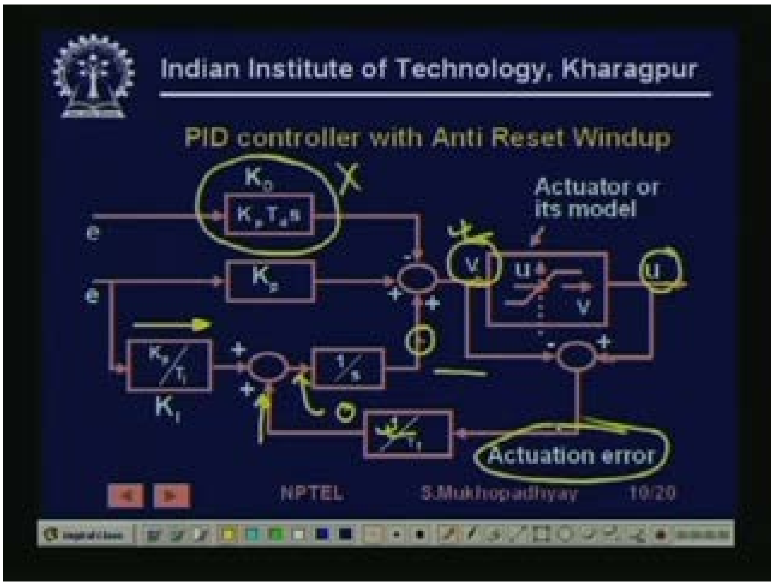


And it happens essentially because of the fact that see during the time when the error is persisting say, from this point the error is persisting. So, the proportional input is remaining constant, but the, but the integral input is growing. So, suppose at this time it has reached this value now the set point is reduced. So, it is at this point then the set point is reduced. Now, what is going to happen and this is the saturated output. So, the integral is way beyond the saturated level.

So, now that it is the set point is reduced now the error has become negative. So, the integral value is now reducing, but still it is positive see at this point, at this point the control input is still greater than the saturated level. So, what is actually this level and therefore, the output persists. So, only at this time after so much time does it come to the below the saturated input level and then it goes further below.

So, the so from this point onward the output will start with this. So, this is what happens. So, this is. So, the whole idea is that the integral should not be allowed to blow up and continuously blow up with time, if the, if the, sorry if the error persists due to a phenomenon like actuator saturation. So, that is precisely that can be done in many ways.

(Refer Slide Time: 22:09)



And here is, here is one here is a scheme one scheme one of many possible schemes, which will realize that. So, how do we do that? So, look at this controller simple controller we have, we have the usual P D that the derivative term, which you can ignore for the time, for the time being it is not concerned.

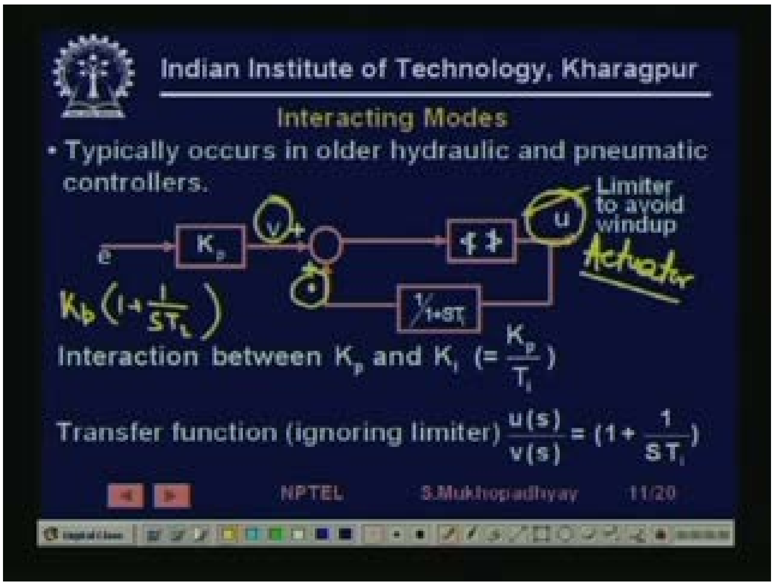
So, we have a proportional term, which is coming you also have a derivative term, which is coming, which we need not consider at the in this slide. So, what is happening here that here actually it is here, suppose this is the actuator right. So, the actuator has a saturation characteristic. So, even if this is the controller output and this is the plant input in between.

So, what you are doing is you are actually sensing the physical plant input you could either do that or you could have an have a model of the actuator in the controller itself and check before giving the input check whether this is really going to cross the actuator limit. So, you can either do it in software or you can use the sensor to again see what input is going.

Now, when V becomes larger than U then you are, and then this actuation error becomes negative. Now, what you want to do is now you take this actuation error and you feed it. So, here a negative term is coming, and here error is positive. So, through the PID integral term a positive term is coming. So, you have to define this gain in a suitable manner such that whenever V this becomes negative, this signal becomes 0, this signal becomes 0. So, when this signal becomes 0 this integrator does not build up. So, this integrator output remains at constant value.

So, you see that whenever you are giving an input, which is going to cause an actuator saturation the this special path, which we have added to the PID controller will now prevent, will now prevent the integral term from blowing up so that when the set point is reduced the plant output will follow very smoothly right. So, this is the scheme, this is one of the schemes, which can be used for anti-reset windup; sometimes integral windup is called also called reset windup. So, coming to the next one, now as I was telling that PID controllers were historically many of them were made using hydraulic and pneumatic devices. So, they use to have you know certain realization structures.

(Refer Slide Time: 24:57)



So, this is a typical structure here where you know as I, as I said that you know one part, one part of the controller used to be realize, use to realize the gain typically you know devices like flapper nozzles, which we will see. There are, there are various kinds you know bellows or orifices constrictions, which are used to realize this time constant. So, the controller structure look at this structure.

So, if you realize this for the time being let us forget about this, one assume that it is it goes directly it is the one. So, if you take this structures then you will find that, you will find that what is the if you compute the transfer function between U and V. You compute the transfer function between U and V, first of all note that there is that is K i is realizes K p by T i.

So, there is interaction, this is called an interactive mode because if you change the proportional gain K p, then the integral gain K i also changes. So, whenever you change K p, if you want to keep the integral gain constant you have to also change T i. So, the various parameters cannot be varied in a non interactive mode, but they must, they will be interactive, and the transfer function between V and U ignoring the limiter. This is the limiter that is if the value goes beyond the certain value, it will limited, if it goes below a certain value it is also limited. If, we ignore the limiter for the time being and you will find that the transfer function between U and V is given as 1 plus 1 by S T i. So, when you multiply it by K p, you get the transfer function of a P I controller.

So, you can also see. So, basically what you have realized is the same transfer function that is K p into 1 plus 1 by S T i, but you have realized it in this way. Now, if you now let us look at the role of the limiter, which is also used in this structure to avoid integral windup. So, you see that if there is, if there is, if there is if U goes to high this is actually going to the actuator. This is the controller output, which is going to the actuator.

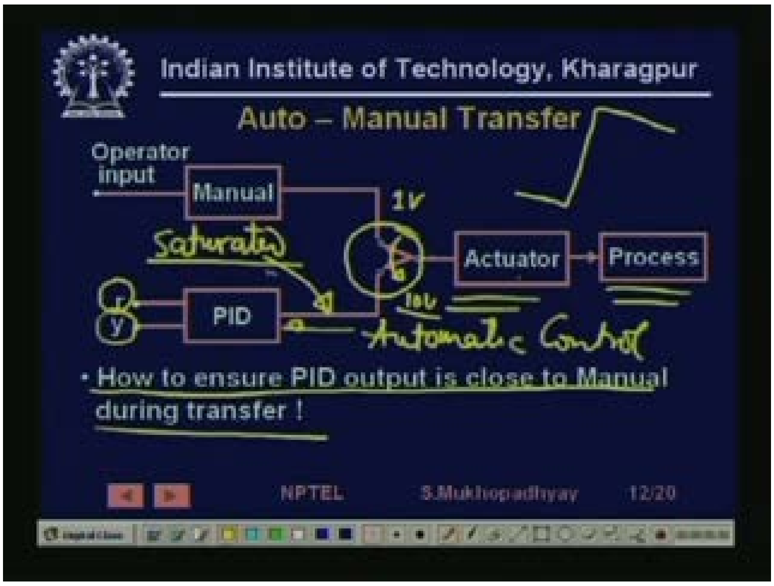
So, if this U goes to goes very high, then what is going to happen is that this limiter, which is inside the controller itself is going to, is going to limit this. So, this U will become constant. So, when this U becomes constant you can see these are, these are simple first order transfer function at that point of time this input will also be constant

So, now what is happening is that the error is the error is constant. So, therefore, this V is constant, and because this U has gone to high level. So, even at some level depending on the where you have set the limiter this is also constant. So, therefore, U becomes constant. So, the output of the P I controller does not build up indefinitely, but gets limited

So, this is another way by which an anti reset windup scheme can be implemented typically in hydraulic and pneumatic controllers. Now, we will look at a phenomenon another problem with occurs typically with integral control and that happens when you have you know auto manual transfer.

Now, let me first explain this term. There are, there are, there are many most processes will also allow the operated to give input that is if he, if he wants in then certain situations you can bypass the automatic controller. And rather using some, using some input device like a, like a, like a potentiometer or a knob or a switch you can given manual input to the plant and you can slowly build it up. Then, at may be some purpose right or and then but then finally, you do not want to run the run the plant manually all the time. So, want to switch over to the automatic control.

(Refer Slide Time: 29:26)



Now, during this switch over problems can occur as we will see here. So, here is a case where so you see that this is a process, this is the actuator and the input to the actuator can come either from the automated PID controller. So, this is automatic control automatic controller and this is manual. So, the operator is actually giving some input here, and here is a switch, here is the switch, which you can flick. So, that the actuator gets its, gets its input either from the PID controller or from the manual controller.

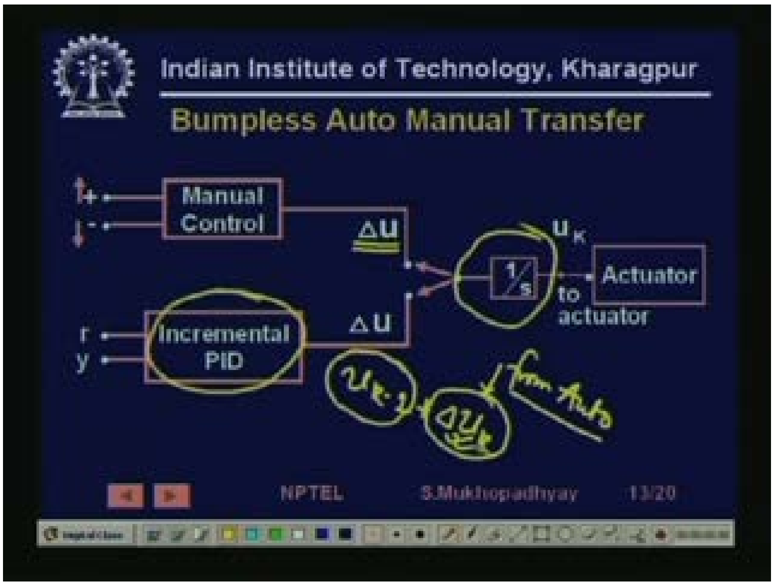
Now, imagine now the main question is that when I am transferring, how do I know? For example, suppose here the input was let us say one volt, now how do I know that when I flick this switch to auto I shall, I shall also here the input existing may be 10 volts. So, now what will happen that previously the actuator was in, was in 1 volt was getting an input of 1 volt. Now, from one volts suddenly a 10 volt output will suddenly a 10 volt will go to the actuator.

So, the actuator if it is a motor or if it is a valve, it will it might get a shock. Similarly, the process also will get a very will tend the get very high input. So, this shock that is we normally try to operate the processes. Then, we if we want to increase the input. So, will wrap it up gradually you do not give an input 1 volt. Now, 10 volt then again minus 2 volts so such inputs are sometimes detrimental to the equipment either in the process or to the, or to the actuator equipment.

So, the question is, question is how to ensure that the PID output is close to the manual during transfer. In fact it often it is not because of the fact that the PID control remember that its output is not going to the actuator, but is all the time getting both the set point and the measurement.

So, it is all the time computing the error computing its integral everything it is doing. So, it is quite likely that the PID control output is actually saturated during the time that you are manipulating the process with manual control. It is quite possible that the PID output has got saturated. It is start either at its negative maximum or at its positive maximum. So, now if you, if you suddenly flick the switch over you are likely to give the plant a shock and we want to avoid that. So, how to avoid that, so for avoiding that for avoiding that we rather than giving the input U, we would like to give the input delta U.

(Refer Slide Time: 32:28)



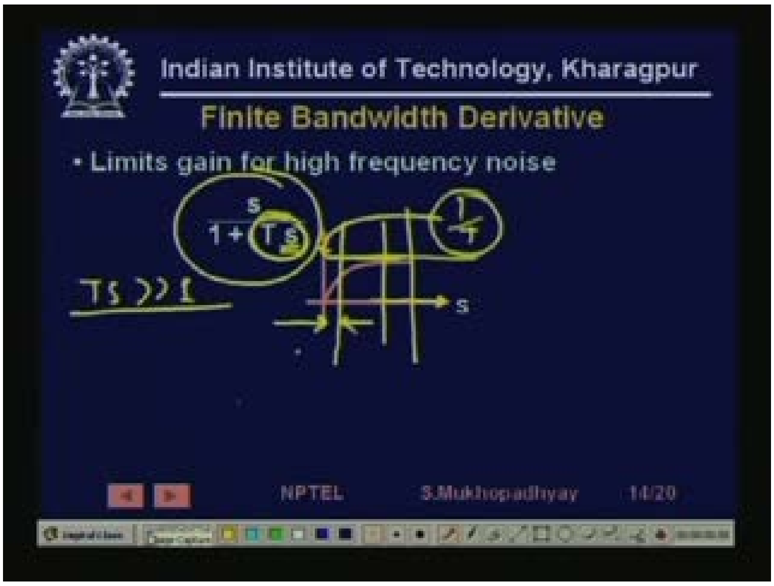
So, you see that this is a clever scheme, which avoids that avoid that process. So, if suppose you are in manual process. So, you are every time you are actually giving delta e. So, you are maybe there is a plus minus switch, and you are flicking the switch to plus. So, the input is going up every time you are giving positive delta e. If, you are flicking the switch to minus you are giving negative delta e, and these this delta U are getting integrated here by some device may be in the actuator.

Now, and the PID controller is also does not give you U, but does give you incremental input delta e every time it computes delta e, how it computes delta U we will see very soon. But suppose the PID control is implemented in a form, which is known as the incremental PID form and its gives you delta U right.

So, now what is going to happen? Now, suppose up to U k minus 1, U were in manual now suddenly in U k just between U k minus 1 to q between the time instances k minus 1 to k. You have flicked the switch to on so what will happen is that now a U k minus 1 plus delta U k will term will be added to the actuator. This is will come from what this will come from the auto, but you see that this actually a delta term. So, it is an increment. So, it cannot be very large.

So, the process will get the old manual input plus a little change, which is due to auto. So, it will not get up and will slowly take the process from one input to another input. So, the transfer from auto to manual mode is going to be bump less that is the terminology, which is used. So, these are a shock. So, gradually these delta U k will build some problems, which you need to you know, you know I mean take care of when you are trying to implement especially when you trying to implement integral term. Now, let us come to the derivative term. So, let us look at the first problem with derivatives that is that that derivatives.

(Refer Slide Time: 34:50)

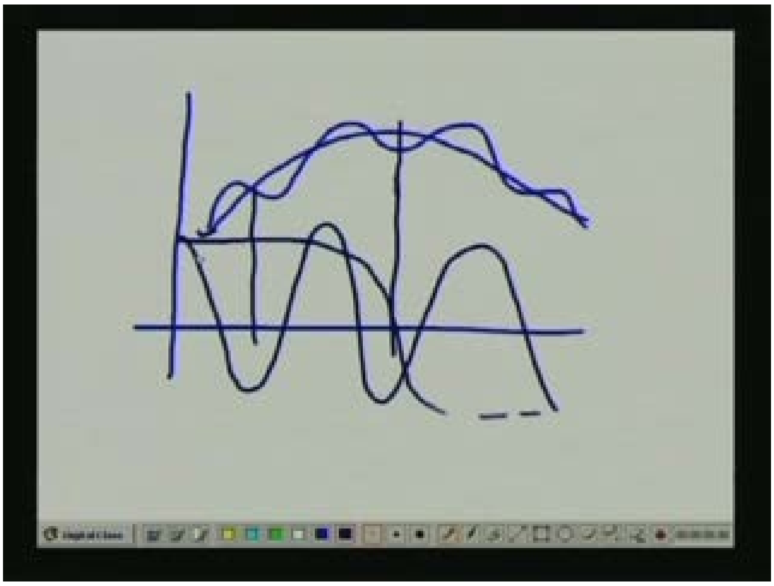


I am sorry that derivatives typically tend to blow up high frequency noise, and where does high frequency noise come from high frequency noise comes from sensors amongst one thing. For example, typically let us consider some you know a flow sensor you know flow sensor flow is always turbulent.

So, you know for the fluid actually flows in random fashion whenever you have a slope beyond a certain velocity flow is turbulent. So, whatever sensor you now the turbulence induces frequency, which are much higher than the average volumetric flow rate. So, while the average volumetric flow rate may be varying like this the signal that you will get from the sensor may induce very high frequency components.

So, the signal these signals are actually due to, due to the turbulence. So, this is, this gives a noise right. Now, the point is that if you take this signal and if you suppose, if you consider the case that consider the case that you have a this point needs to be understood.

(Refer Slide Time: 36:05)



Suppose, you have a signal, which goes like this and you on this you have a noise, I am drawing actually noise will be much more higher frequency. Now, if you take derivative of this signal, what will happen? The derivative of the derivative of the low frequency original signal will remain will remain positive up to this point will slowly fall and then will become negative here. The derivative around this is going to be positive then it will fall to 0 and then become negative, what happens to the derivative of the other signal.

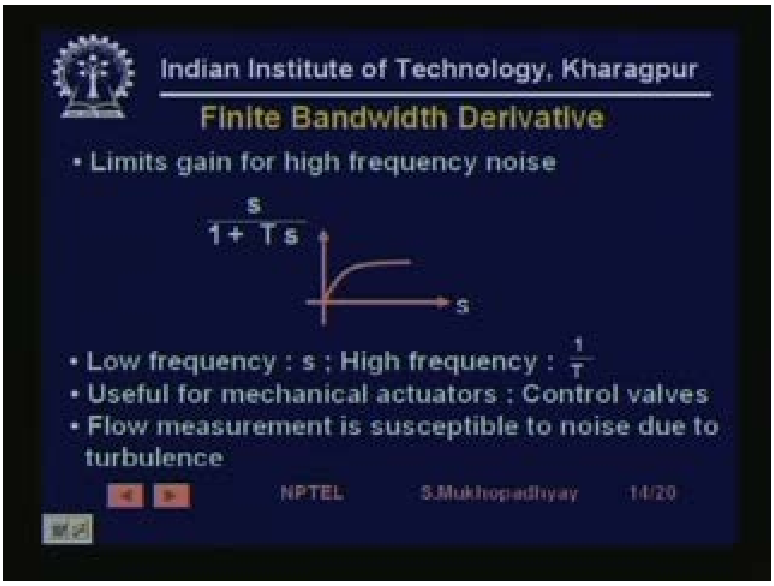
See the derivative of the other signal is going to start from positive with within this short time it will reach a negative maximum, and then it will again. So, it will be, it will be widely varying. So, you see that while these signals are more or less close to each their derivative terms are completely different.

So, if you calculate a derivative exact pure derivative then even a small amount of high frequency noise is going to give you a lot of difference in the control input. So, therefore, you need to use a derivative, which will act like a derivative up to a certain frequency, but beyond a certain frequency its gain will not be, will not blow up right. So, we need to limit the gain for high frequency noise.

So, to do that we simply the transfer function of a derivative is S, if we now consider the transfer function of the, of the signal S by 1 plus T s imagine. This side I need to change the pen. So, that what is the gain of the. So, varies that is the as the frequency varies S by 1 plus T s will at low frequencies this term is small compared to 1.

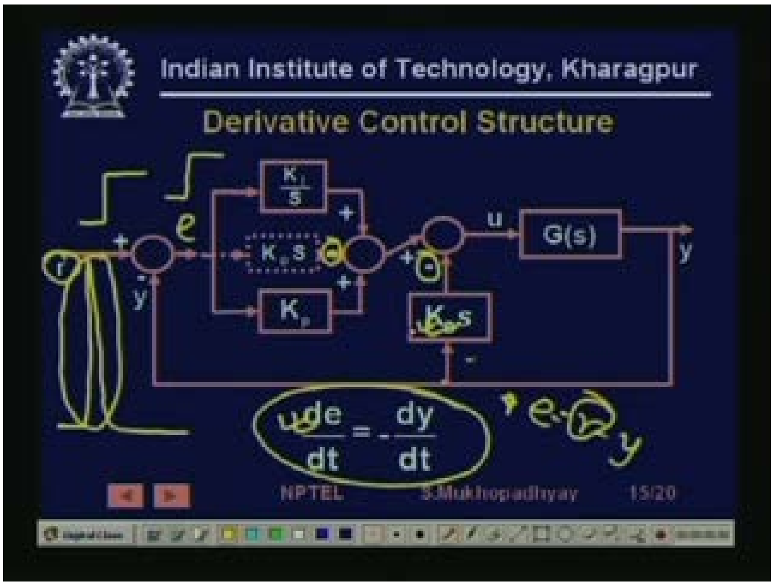
So, it acts, so it acts like, so it acts like yes in this part of the region, while in this part of the region where S is large. Then, it acts like this then its T s is, T s is much greater than one and therefore, this gain becomes equal to 1 by T. So, this level is 1 by T. So, you see that we have in the lower frequency region we are having a derivative, but in the higher frequency region we are having a constant gain of 1 by T. So, that is what so that is how you need to you need to realize your derivative.

(Refer Slide Time: 39:20)



So, the idea is that for low frequency it is S for high frequency, it is 1 by T. Now, this is also useful for some kinds of mechanical actuator as I said that if you give a very high frequency signal, and then creates derivative. Then, you are going to give it very high frequency positive and negative torques, which is not good for a mechanical actuator, like control wall that it might damage the wall. As I have given an example that such noise may come from various kinds of measurements especially flow measurements, which are very common in a, in a industrial process.

(Refer Slide Time: 40:01)



So, now we come to the now there is, there is, there is a second problem associated with the derivative control structure. So, far as we have seen we have implemented, we have we have implemented the PID control like this that is the error term, this is the error term. A error terms goes here gets multiplied by K p it is multiplied by K i and gets multiplied by K D also.

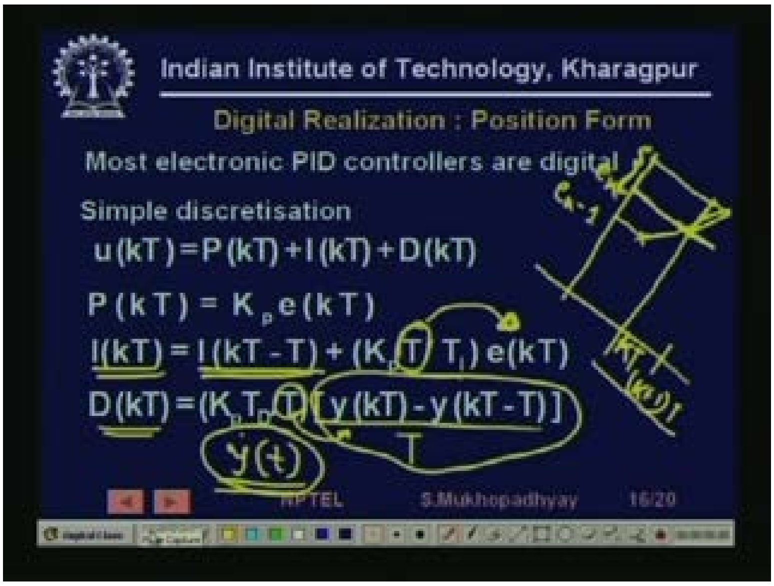
Now, as we have seen that we want to avoid giving shocks to the process right. So, what happens when you in many cases the set point is changed like a step. So, if you change the step, what is going to happen to the error, it is also going to change like a step. So, then what will happen to the derivative output here.

It is going to be at this time, it is going to rise very much and then it is going to fall to 0. So, you are going to give a shocking input to the plant, if you put the derivative here. So, now the question is that how can I avoid such shocks, but if the if r is not changed, if when r is constant. If, there is, if y changes due to various other factors like disturbances we I need to keep the derivative control. I do not want to sacrifice derivative further want to have it except for the instance, where r changes like a step.

So, clever we are doing that is by realizing that when R is constant d e by r is equal to rather e is equal to r minus y. So, d e by d t is equal to when r is constant it is minus of d y by d t. So, rather than having d e by d t here I can also take y here, and then change the sign and implement the same block here. So, when r is not changing I am going to get the same effect as d e by d t, but when r is, r goes as, goes to a step there is, there is absolutely no effect here. So, during that time it will, it will simply slowly rise and corresponding effect. You will get as y rises, you will again getting get the P D effect, but the shock will not come.

So, this is something, which is to remembered when a derivative control is to be implemented. Now, we come to the last topic that is the digital realization because now a day most controllers are actually implemented using microprocessor. So, how do we implement the PID equation in a, in a in microprocessor? So, that is very simple.

(Refer Slide Time: 42:44)



That is we do at, we do what is known as a discretisation, in other words the integral, we simply replace by a sum. So, we say that in a digital controller we can compute inputs at certain instance of time. So, if T is that those instance of time are called sampling instance. So, at the K th sampling instance where the time value is equal to K of T K into T.

The input is again consists of three terms that is the proportional value at K T, the integral value at K T and the derivative value at K T. Now, the question is how do we compute this proportional integral, and derivative terms? So, the proportional controller is simply P k T is equal to K p into e K T. So, we just sample the equation. The integral controller is the integral is actually realized by a what is known as a trapezoidal integration right not even not even trapezoidal. This is, this is you know, what is called a backward difference integration right.

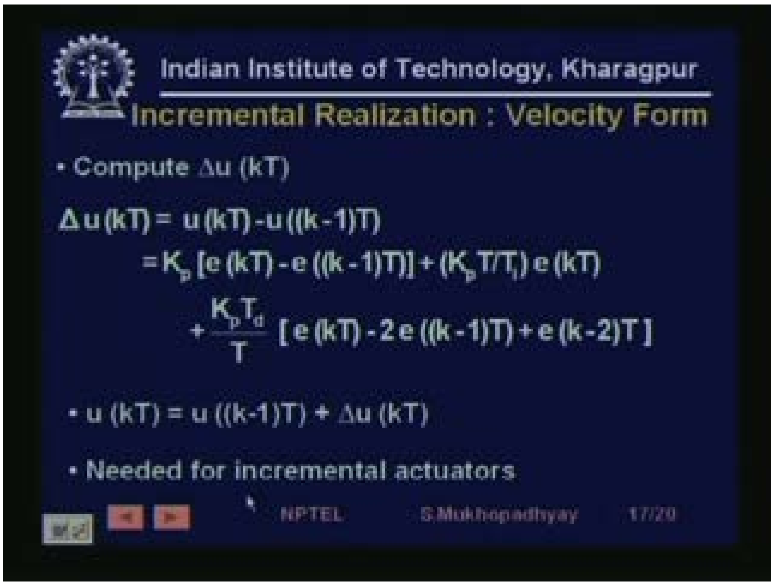
So, what we are doing is that we simply assume that since e K T is going to be constant over that over the time interval K minus 1 to K. So, the integral. So, we want to actually what. So, what we are doing is that basically we are doing that simple thing that if this is e K T, this is e K minus 1. This is e K then after we get e K, what is going to be the integral and this e K plus 1 and the error is decreasing like this. So, between e K and e K to e K minus 1 that is between rather K T to K plus 1 T.

I assume that this error is going to be maintained. So, the integral will rise simply as a, as a as rectangle. So, I write, so I multiply this integral, I realize by multiplying e K T by T that will be the integral, and then I add it with my previous value of the integral to get the present value of integral.

Similarly, the derivative term we make a simple bring this T here. So, then you get this is an, this is a basically an approximation of y dot t. So, basically I construct, I have to construct approximations of the derivative of y and integrals of y using samples that is what we do. And this implementation from is often called a position form. So, it is called a position form because the whole input U is U is calculated.

Now, as we have seen that in some cases it is, it is rather necessary to generate delta U rather than U as we have seen right. So, then and now the question is that how do we generate delta e that is called an incremental realization of the PID controller or sometimes called a velocity form.

(Refer Slide Time: 46:15)



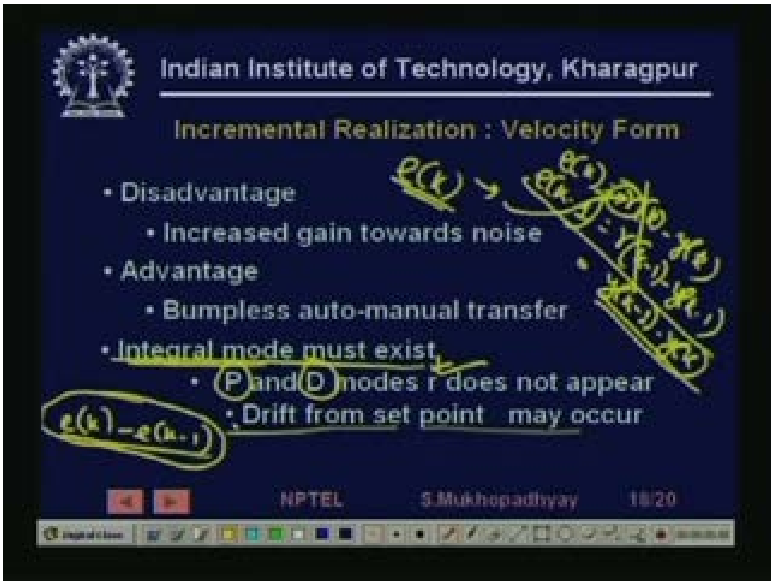
So, what happens in the velocity form we need to compute delta u k T. So, delta u k T is very simply computed delta u k T is nothing but u k T minus u k minus ` T. So, we simply substitute the previous formulae of u k T and u k minus 1 and then subtract. So, what will happen is that see the what happens is the proportional term subtraction gives this K p into e k T minus e k minus 1 T.

Similarly, the integral term is basically, I k T minus I k minus 1 T. So, it will give that additional term which I obtained. So, this is the integral part of delta u, and the derivative part of delta u is this which is basically the difference between this and the same thing at k minus 1. So, you basically by taking a difference of these terms you can generate the generate delta u very simply. So, it is not a difficult problem, and then finally, we have to give u we cannot give to the plant. Finally, we have to give u, So, we simply add u k minus delta u k T, we have computed.

So, we have to simply add u k minus 1 T with that, and there are also some kinds of actuators where let us say like you know step motors where the actuator itself integrates the output. So, this summation you need not, you need not give you just give keep on giving the delta u, and the, and the actuator will gradually move and will continuously add it.

So, in a let us say in a, in a position control using some using some, let us say stepper motor it is, it is very convenient to give the delta u. In fact it is the delta u, which have to be given and. So, therefore, the PID control has been implemented in this form.

(Refer Slide Time: 48:39)



So, as we have seen that there are some disadvantages of this I mean the incremental realization because of the, because of the sample realization. And because we in this case we are actually making a second order derivative not a first order derivative, because there is already a derivative term in U, and we are making another derivative, because we want to compute delta e.

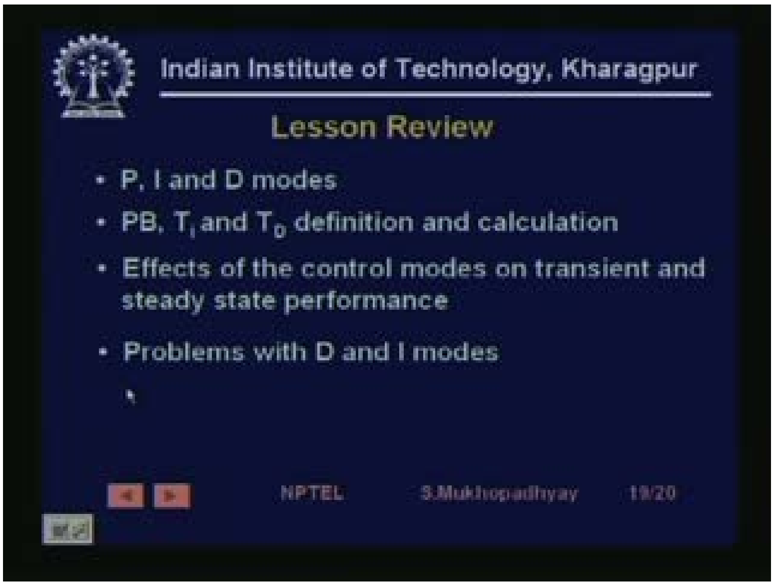
So, because of sampling approximations a high amount of noise may be introduced there that needs to be taken care of. So, your sampling intervals should be good small enough, and the advantages as we have seen is that is very simply possible to give a bump less auto manual transfer and also for some actuators, it is what needs to be given.

There is an interesting phenomenon that happens is that sometimes generally this mode is implemented in this form that is the PID controller when there is an integral mode. Basically, because of the fact that if you see the delta u term you will find that one component contains e k minus, e k minus 1 that is the proportional component. Now, in this component there is no r. There is no that is the there is a reference input does not appear because e k is equal r k minus y k.

Similarly, e k minus 1 is equal to r k minus 1 minus y k minus 1. So, if you subtract and if r is remaining constant then these will get constant, and you will simply get y k minus, y k minus 1 minus y k. So, and the same thing happens to the D mode. So, the P and in the P and D modes r does not appear. So, the reference input. So, even if the in you know even if the output drifts slowly even if because r does not appear so it will. So, the total value of error does not come is not reflected in the, in the control input, but in the integral terms there is an e k term just simply k.

So, that contains r k, that contains r k, the there r k is not canceled. So, it is of, so it is said that generally when you implement digital controllers in incremental form you should have an integral mode otherwise the process may slowly drift from set point without the controller taking corrective action. So, having said that. So, that brings us to the end of our lesson, let us review the lesson once.

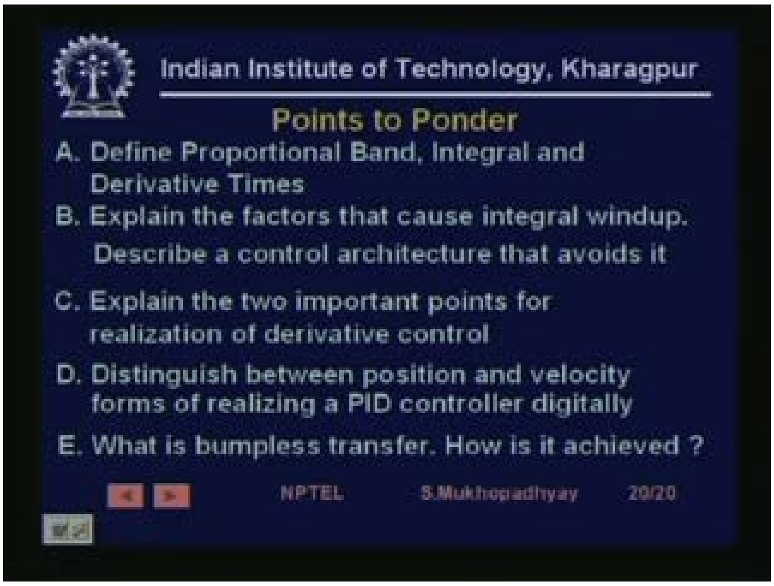
(Refer Slide Time: 51:29)



So, we looked at the P I and D modes and looked at the various terms their definitions proportional bands integral time and derivative time. Then, meaning, how they can be measured so and so. Then, we saw that the these various controlled modes can give you various kinds of transient and steady state performance. For example, we have seen that the integral control mode can cause integral windup they can cause integral windups.

Similarly, we have seen that the derivative control mode can give you a lot of noisy performance, if you have sensor noise. So, some effects of these control modes on performance are discussed that is these problems with D and I modes we have discussed, and finally, we have discussed also a digital realization. Now, next look at some you know points to ponder whose answer again just like earlier lessons we will find within this presentation itself if you look carefully.

(Refer Slide Time: 52:50)



So, you might like to write yourself the definition of proportional bands integral derivative times. It is you would like to explain the factors that cause integral windup. So, integral windup is basically caused by basically caused by 2 factors. So, what are those factors and we have discussed in this lecture 2 control architectures, which will avoid integral windup. So, how to avoid that and then we have seen that while you are implementing derivative control you have to take care of 2 points such that you do not add unnecessary disturbances, and shocks to the plant.

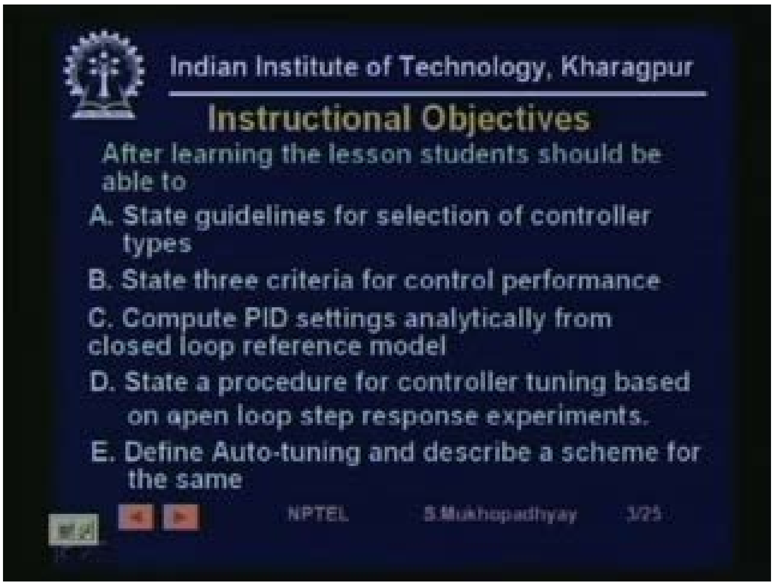
So, what are they and finally, you have seen that a PID controller may be realize in what is known as the position form and a velocity form. So, you need to think how to distinguish between them when which one is required, and also what is a bump less transfer and how it is achieved? So, the answers to these questions are exist in many text books and also within the within this lecture.

Thank you very much today.

(Refer Slide Time: 54:28)

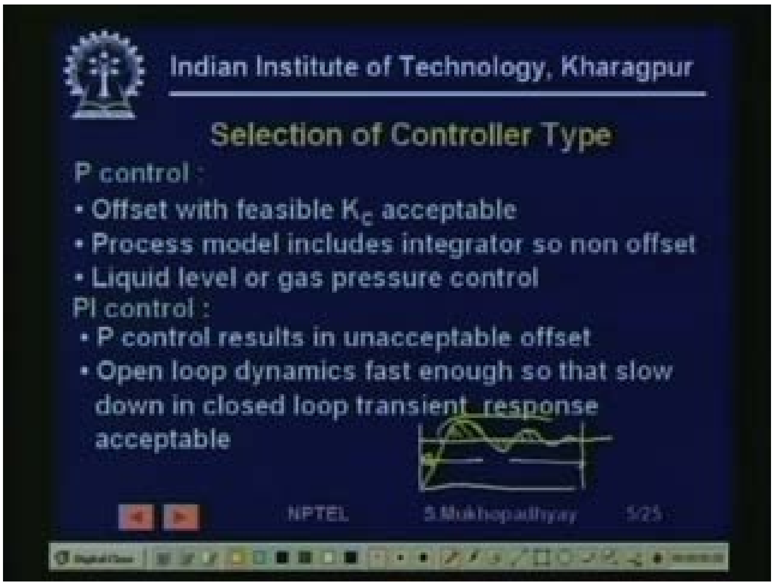
Good morning. So, today we are going to have lesson 13 of the course, which is on PID controller tuning. As we will see that controller tuning is a very important phase of the overall controller design, and it very critically determines the performance of the control loop, which in turn affects the overall quality of the product effects costs. So, it is a very important method to be learnt in the, in the overall context of industrial automation. So, before we get into the business proper, let us first see what are the instructional objectives of the code of the lessons as is the usual practice.

(Refer Slide Time: 55:41)



So, as we state in every lesson after learning the lesson the students should be able to firstly state guidelines for selection of controller types, when I said PID controller. I actually mean a class that is a 3 classes of controller that is P control PI control and PID control, which are most often used in the context of industrial automation. So, the student should be able to select one of these types.

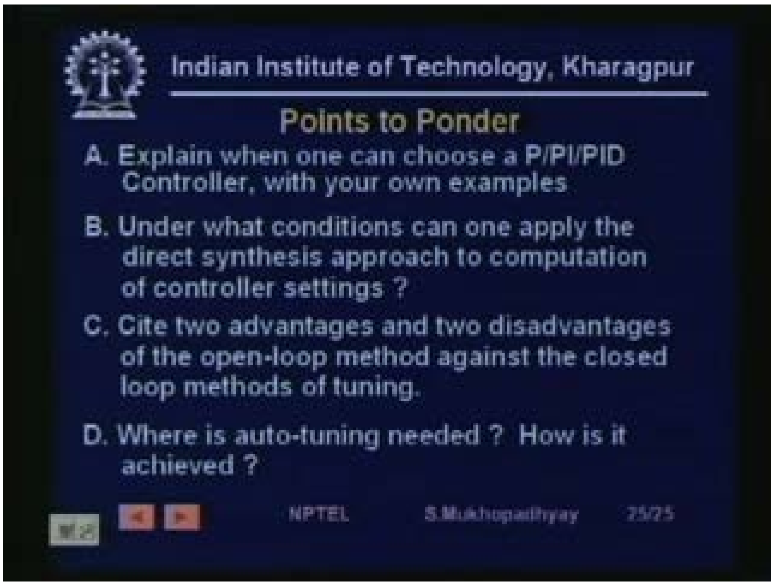
(Refer Slide Time: 56:19)



Then, you can say then even if during this period I have some error and my control performance is may be affected depending on whether I have a, if I have an integral error criteria. Then, it will be less affected, because this positive error is going to cancel this negative error, but even then I am able to tolerate this much of error because of because it is going to die down reasonably fast for me, and then it will stay on for long time right.

So, in such a case, so in other words the this interval is actually small, which means that the, that the open loop dynamics is actually fast enough. It is fast enough compared to what compare to the frequency of set point change. So, that the closed loop transient response is actually accepted. Typical example is flow control points to ponder.

(Refer Slide Time: 57:16)



For you or is that you can try to find out in what situations P, PI, and PID controller are to be use all the answers are in this lecture only, and you can try to find out your own examples of processes. Then, under what conditions you can find under what conditions one can apply a direct synthesis procedure, let say in your application process. You can apply or not or if you can apply why you can apply or why you cannot apply all these things.

You can also cite two advantages and disadvantages of open-loop method against the closed loop method. And finally, you can, you can find under what situation an auto tuning future will be needed, and how is it achieved? How is it, how an auto tuning is achieved one procedure is already given in the lecture. So, here we end.

Thank you very much.